

JP06106810 A
METHOD FOR FORMING IMAGE AND ITS DEVICE
CANON INC

Abstract:

PURPOSE: To form a form overlay image speedily by storing form information which is previously compressed, in cache memory, and synthesizing the stored form information with other image data to form an image, if 'form overlay' is instructed.

CONSTITUTION: Form information is developed in a bit map and compressed, and the compressed form information is stored in work memory 103. If 'overlay' with the form information is instructed, a means 107 for drawing an image reads and stretches the stored form information, and overlaps the beam image of the image data to develop the overlapped image in the bit map, so that the means 107 is activated to draw an image. Consequently, it is possible to draw a form overlay image rapidly.

Inventor(s):

OTANI MIYUKI

Application No. 04254577 JP04254577 JP, **Filed** 19920924, **A1 Published** 19940419

Original IPC(1-7): B41J02100
G06F00312

Patents Citing This One No US, EP, or WO patent/search reports have cited this patent.

(11)特許出願公開番号

特開平6-106810

(43)公開日 平成6年(1994)4月19日

(51) Int.Cl.⁶

識別記号

庁内整理番号

FI

技術表示箇所

B 4 1 J 21/00

A 8703-2C

G O 6 F 3/12

F

審査請求 未請求 請求項の数 4 (全 15 頁)

(21)出願番号

特願平4-254577

(22)出願日

平成4年(1992)9月24日

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 大谷 幸

東京都大田区下丸子3丁目30番2号 キヤ

ノン株式会社内

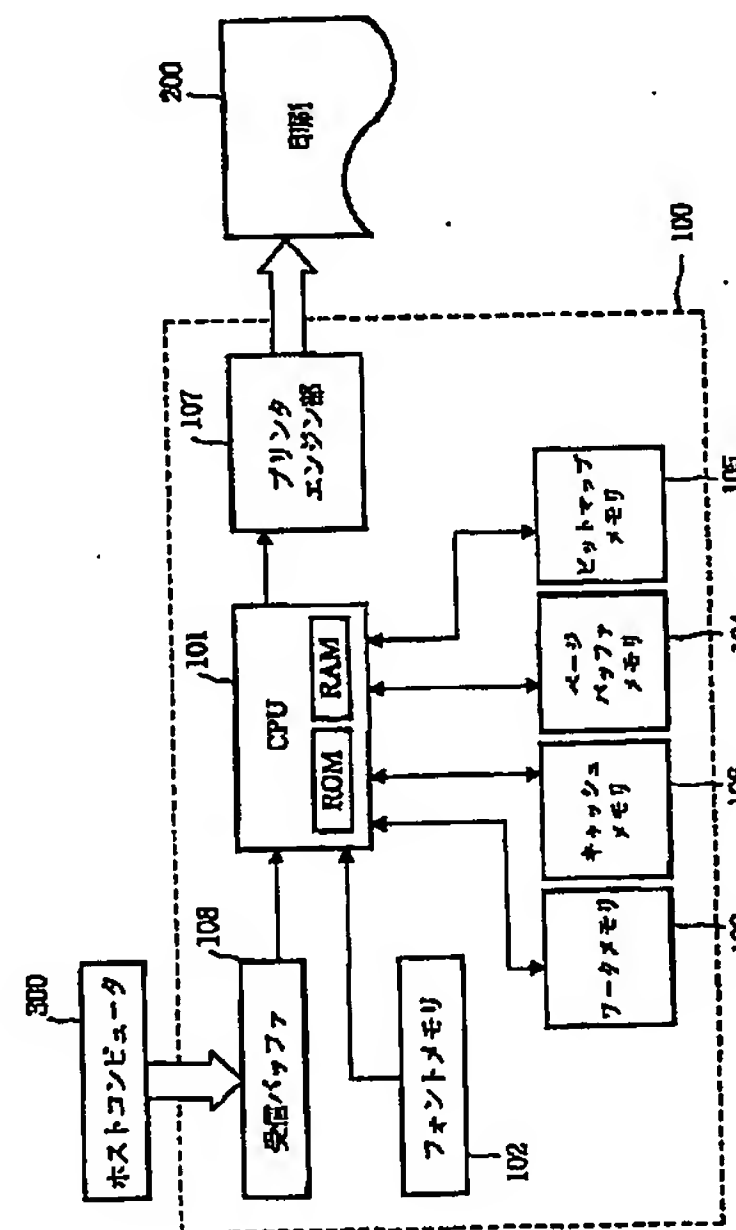
(74)代理人 弁理士 大塚 康德 (外1名)

(54)【発明の名称】 画像形成方法及び装置

(57) 【要約】

【目的】 フォーム情報を圧縮してキャッシュとして記憶しておき、フォームオーバーレイが指示された時に、その他の画像データと記憶されているフォーム情報とを合成して画像を形成することにより、高速にフォームオーバーレイ画像を形成できる画像形成方法及び装置を提供することを目的とする。

【構成】 フォーム情報をビットマップに展開して圧縮し、その圧縮されたフォーム情報をワークメモリ103に記憶しておく。そのフォーム情報とのオーバーレイが指示されると、記憶されているフォーム情報を読出して伸長し、画像データのビットイメージと重畳してビットマップ展開し、画像イメージを作成するように動作する。



【特許請求の範囲】

【請求項1】 定型的なフォーム情報を登録しておき、可変データと重ね合わせて画像を形成する画像形成装置において、
フォーム情報をビットマップに展開して圧縮する圧縮手段と、
前記圧縮手段により圧縮されたフォーム情報を記憶する記憶手段と、
前記記憶手段に記憶されたフォーム情報を読出して伸長し、前記可変データのビットイメージと重畳して画像イメージを作成する画像作成手段と、
を有することを特徴とする画像形成装置。

【請求項2】 ビットマップ展開されたデータを保存するメモリを有する画像形成装置であって、
前記メモリにデータを優先順位を付して記憶する記憶手段と、
前記メモリの残量が所定量以上かどうかを判断する判断手段と、
前記判断手段により残量が所定量以下であると判断されると、前記優先順位に応じて前記メモリに記憶されているデータを消去する消去手段と、
を有することを特徴とする画像形成装置。

【請求項3】 定型的なフォーム情報を登録しておき、可変データと重ね合わせて画像を形成する画像形成方法であって、
フォーム情報をビットマップに展開して圧縮する工程と、
その圧縮されたフォーム情報を記憶する工程と、
フォーム情報と可変データとの重ね合わせが指示されると、その記憶されているフォーム情報を読出して伸長し、可変データのビットイメージと重畳して画像イメージを作成する工程と、
を有することを特徴とする画像形成方法。

【請求項4】 ビットマップ展開されたデータを保存するメモリを有する画像形成方法であって、
メモリにデータを優先順位を付して記憶する工程と、
メモリへデータを登録する際、そのメモリの残量が所定量以上かどうかを判断する工程と、
その残量が所定量以下であると判断されると、記憶されている優先順位に応じて前記メモリに記憶されているデータを消去する工程と、
を有することを特徴とする画像形成方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、例えばホストコンピュータ等の外部機器からの画像情報を入力して画像を形成する画像形成方法及び装置に関し、特に定型的なフォーム情報を1つ又は複数登録しておき、指定されたフォーム情報を指定されたページに、その他のデータと重ね合わせて形成する画像形成方法及び装置に関するものである。

る。

【0002】

【従来の技術】 近年、レーザビームプリンタなどの印刷装置では、内蔵フォントデータとしてアウトラインフォントが採用されている。このアウトラインフォントの特徴としては、加工の容易性、印刷した文字の鮮明さ等にあるが、文字パターンを発生するまでに要する時間がドットパターンを用いる場合よりも長くなるという欠点がある。そこで、1度パターンに展開されたドットイメージをキャッシュメモリに記憶しておき、再び同じサイズの同じ文字コードが入力されると、そのキャッシュメモリより、既にパターン展開されているドットイメージを読出して印刷するという手法が用いられている。

【0003】 また、このような印刷装置では、ホストコンピュータから送られてきた定型的なフォーム情報をコード情報で記憶しておき、そのフォーム情報と重ね合わせて印刷するように指示されると、その印刷の際にそのフォーム情報をイメージに展開していた。

【0004】

【発明が解決しようとする課題】 一般に、前述したフォーム情報は、連続したページに展開されるものであるため、上記従来例のように各ページ毎に展開処理を行うと、その展開処理に多くの時間を費やしてしまうことが問題となっていた。特に、最近スケーラブルフォントや図形情報などのラスライジングに時間のかかるフォーム情報が増大しており、展開時間の短縮に対する要望が拡大している。

【0005】 また、前者のキャッシュメモリを使用する場合では、キャッシュメモリが一杯になり、次の文字が展開できなくなると、そのページで使用されていない特殊文字のパターンをキャッシュメモリより除去し、それでもキャッシュメモリの容量が不足している時はそのページで使用されていない文字パターンを全て削除して文字を展開する。このため、通常 of 文書データを大量に印刷している時には、せっかくキャッシュメモリに文字パターンを蓄積していても、それら文字パターンデータがキャッシュメモリ一杯になる度にクリアされてしまい、印刷速度を向上することができなかった。

【0006】 本発明は上記従来例に鑑みてなされたもので、フォーム情報を圧縮してキャッシュメモリに記憶しておき、フォームオーバーレイが指示された時に、その他の画像データと記憶されているフォーム情報とを合成して画像を形成することにより、高速にフォームオーバーレイ画像を形成できる画像形成方法及び装置を提供することを目的とする。

【0007】 更に、本発明は、キャッシュメモリの登録データに優先順位を持たせ、キャッシュメモリの残量が少なくなると、その優先順位の低い順に登録されているデータを消去するようにしてキャッシュメモリの使用効率を高めるようにした画像形成方法及び装置を提供する。

ことを目的とする。

【0008】

【課題を解決するための手段】上記目的を達成するために本発明の画像形成装置は以下のような構成を備える。即ち、定型的なフォーム情報を登録しておき、可変データと重ね合わせて画像を形成する画像形成装置において、前記フォーム情報をビットマップに展開して圧縮する圧縮手段と、前記圧縮手段により圧縮されたフォーム情報を記憶する記憶手段と、前記記憶手段に記憶されたフォーム情報を読出して伸長し、前記可変データのビットイメージと重畳して画像イメージを作成する画像作成手段とを有する。上記目的を達成するために本発明の画像形成方法は以下のような工程を備える。即ち、ビットマップ展開されたデータを保存するメモリを有する画像形成方法であって、メモリにデータを優先順位を付して記憶する工程と、メモリへデータを登録する際、そのメモリの残量が所定量以上かどうかを判断する工程と、その残量が所定量以下であると判断されると、記憶されている優先順位に応じて前記メモリに記憶されているデータを消去する工程とを有する。

【0009】

【作用】以上の構成において、フォーム情報をビットマップに展開して圧縮し、その圧縮されたフォーム情報を記憶しておく。そのフォーム情報とのオーバーレイが指示されると、記憶されているフォーム情報を読出して伸長し、可変データのビットイメージと重畳して画像イメージを作成するように動作する。

【0010】

【実施例】以下、添付図面を参照して本発明の好適な実施例を詳細に説明する。尚、以下の説明においては、印刷装置の場合で本発明の実施例を説明するが、本発明はこのような印刷装置に限定されるものでなく、例えばディスプレイ装置やディスプレイを備えたコンピュータ装置等にも適用できることはもちろんである。

＜装置構成の説明（第1図）＞第1図は、本発明の第1実施例の印刷装置の構成概略を示すブロック図である。

【0011】図1において、100は本実施例の印刷（プリンタ）装置であり、この印刷装置100により印刷された結果を200で示している。300はホストコンピュータで、実施例の印刷装置100に印刷データを伝送している。

【0012】次に、この印刷装置100の構成を説明する。

【0013】108は受信バッファで、ホストコンピュータ300から伝送されてくる印刷データを一時的に保存している。102はフォントメモリで、アウトラインフォント等の文字フォントデータを文字コードに対応して記憶している。103はフォーム情報を展開しておくためのワークメモリ、106はフォントキャッシュメモリである。104はページバッファメモリで、受信データ

をページ単位に格納している。105はビットマップメモリで、1ページ分のイメージデータをビットマップに展開する。107はビットマップメモリ105に展開されたイメージデータに基づいて、実際に印刷処理を行うプリンタエンジン部である。

【0014】第2図は第1実施例の印刷装置において印刷するオーバーレイ情報のサンプルであるフォーム情報201と、一般的な文書データ（可変データ）202を示す図である。

【0015】このフォーム情報201は、コマンドコードの状態、予めホストコンピュータ300から送信されてページバッファ104に格納され、オーバーレイ印字の指示がされた時に初めて、ビットマップメモリ105にビットマップのデータとして展開される。そして、一般的な文書データ202等が、このフォーム情報に重ねてビットマップ展開され、そのイメージデータがプリンタエンジン部107に送られ印刷される。

【0016】この時、ビットマップメモリ105において、文書データがビットマップ展開される前に、既にビットマップメモリ105に展開されているフォームデータ201をワークメモリ103に記憶しておく。ワークメモリの容量が足りない場合には圧縮して記憶する。これにより、次にオーバーレイ印刷が指示された時には、ページバッファメモリ104のコードを基にフォーム情報をビットマップ展開することなく、ワークメモリ103に記憶されている圧縮されたフォーム情報を伸長しながらビットマップメモリ105に転送して展開することにより、フォーム情報をオーバーレイしてビットマップ展開する時間を短縮することができる。

【0017】第3図は本発明の第1実施例の印刷装置におけるCPU101のオーバーレイページ出力処理を説明するためのフローチャートである。尚、この実施例では、ホストコンピュータ300よりのデータ受信は、図3に示す処理と並行して行われており、受信バッファ108を介してページバッファメモリ104に印刷データが順次格納されている。また、この処理に先立って各ページに重ね合わせて印刷するための第2図に示すようなフォーム情報201が、予めホストコンピュータ300より伝送されてページバッファメモリ104に記憶されているものとする。

【0018】まずステップS1で、オーバーレイ印刷指示がホストコンピュータ300より入力されると、ワークメモリ103にフォーム情報が圧縮されて記憶されているかどうかを判断する。記憶されていない場合にはステップS3に進み、ページバッファメモリ104に記憶されているコードデータをビットマップ展開してビットマップメモリ105に格納する。次にステップS4に進み、そのビットマップ展開されたデータをビットマップメモリ105より読出して圧縮しながらワークメモリ103に記憶する。

【0019】一方、ステップS1でワークメモリ103にフォーム情報が記憶されていた場合にはステップS2に進み、ワークメモリ103のフォーム情報を伸長しながらビットマップメモリ105に展開する。

【0020】そして、ステップS2或いはステップS4を実行した後ステップS5に進み、そのフォーム情報に重ね合わされる文書データをビットマップメモリ105に展開し、フォーム情報と、一般的な文書データとが重なった状態のビットマップデータ（イメージデータ）を、プリンタエンジン部107に送出して印刷処理を行う。

【0021】上記のような処理を行うことにより、フォーム情報をビットマップ展開する時間を短縮できる。また、ビットマップ展開されたフォーム情報は圧縮して記憶されるので、メモリを大きく占有することもなく、高速かつ効率的な印刷を行うことが可能となる。

【0022】次に、本発明の第2実施例について説明する。

【0023】前述した第1実施例では、フォーム情報を記憶するエリアとしてワークメモリ103を使用する場合について説明したが、この第2実施例では、キャッシュメモリ106aにフォーム情報を記憶する場合について説明する。

【0024】第4図は本発明の第2実施例における印刷装置の概略構成を示すブロック図であり、第1図におけるワークメモリ103を除いたもので、他の構成は前述の図1と同じあるため、それらの説明を省略する。

【0025】このような構成において、キャッシュメモリ106aの容量を大きくとることにより、フォーム情報も他のスケーラブルデータと同様にキャッシュメモリ106aに登録して管理する。尚、キャッシュメモリ106aにおいて、フォーム情報と他のデータとの間の優先順位を設定することにより、より有効にキャッシュメモリ106aを活用することができるようになり、印刷速度をより向上することができる。

【0026】第5図は本発明の第2実施例の印刷装置におけるCPU101のオーバーレイページ出力処理を説明するためのフローチャートである。尚、この実施例では、ホストコンピュータ300よりのデータ受信は、図5に示す処理と並行して行われており、受信バッファ108を介してページバッファメモリ104に印刷データが順次格納されている。このページバッファメモリ104に記憶されている文字コードやフォーム情報を順次読出してビットマップ展開するものとする。

【0027】まずステップS11で、文字コードを読出したかどうかを判断し、文字コードであればステップS12に進み、その文字コードに対応するパターン情報がキャッシュメモリ106aに記憶されているかどうかをみる。キャッシュメモリ106aに登録されていればステップS13に進み、該当する文字パターンをそのキャ

ッシュメモリ106aより読出して、ビットマップメモリ105に展開する。一方、ステップS12で、該当する文字のパターンデータがキャッシュメモリ106aに登録されていない時はステップS14に進み、フォントメモリ102を参照して、その文字コードをビットマップ展開してビットマップメモリ105に記憶する。更にステップS15に進み、その展開した文字パターンをキャッシュメモリ106aに登録する。

【0028】次にステップS16に進み、フォーム情報を読出したかを調べ、フォーム情報であればステップS17に進み、そのフォーム情報がキャッシュメモリ106aに登録されているかどうかを判断する。登録されていればステップS18に進み、キャッシュメモリ106aの圧縮されて登録されているフォーム情報を伸長しながら読出し、ビットマップメモリ105に展開する。一方、ステップS17でキャッシュメモリ106aに登録されていない時はステップS19に進み、そのフォーム情報をイメージに展開してビットマップメモリ105に格納するとともに、ステップS20でそのビットマップ展開されたフォーム情報を圧縮してキャッシュメモリ106aに登録する。

【0029】こうしてステップS21に進み、ページバッファメモリ104の全データのビットマップメモリへの展開が終了したかを調べ、終了していればステップS22に進み、そのイメージデータをプリンタエンジン部107に出力して処理を終了する。

【0030】次に図6を参照して、本発明の第3実施例を説明する。

【0031】前述の第1或いは第2実施例では、ワークメモリ103或いはキャッシュメモリ106aに、フォーム情報を登録できるだけの容量が残存していない場合を考慮していないようにして説明した。この第3実施例では、図6を参照すると明らかなように、例えばワークメモリ103の容量が足りない場合は、圧縮・登録処理を行わないようにしている。尚、キャッシュメモリ106aにフォーム情報を記憶するだけの容量がない場合も同様にして実現できるため、それらの説明を省略する。

【0032】図6は本発明の第3実施例におけるCPU101のオーバーレイページ出力処理手順を説明するためのフローチャートである。前述のフローチャートの場合と同様に、ホストコンピュータ300よりのデータの受信はこの処理と並行して行われており、受信バッファ108を介してページバッファメモリ104に受信データが順次格納されている。

【0033】まず、オーバーレイ印刷指示をホストコンピュータ300から受信すると、ステップS31でワークメモリ103にフォーム情報が記憶されているかどうかを判断する。フォーム情報が記憶されていない場合はステップS32に進み、ビットマップメモリ105にコードデータを展開する。その際に、画像の開始点（最初

のデータ変換点)から終了点(最後のデータ変換点)までのデータサイズ(バイト)をカウントし、ステップS34で予測圧縮データサイズ=(データサイズ×0.8)バイトで示されるメモリ容量がワークメモリ103に残存しているかどうかを判断する。このメモリ容量が残存していない場合にはステップS36に進み、このフォーム情報に重ね合わされる文書データをビットマップメモリ105に展開し、フォーム情報と、この文書データとが重なった状態で完成したビットマップメモリ105のイメージデータを、ステップS37でプリンタエンジン107に送出して印刷を行う。

【0034】一方、ステップS34でワークメモリ103に、予想圧縮データサイズ分の空きエリアがあった場合にはステップS35に進み、ビットマップメモリ105に展開されているフォーム情報を読出し、そのフォーム情報を圧縮しながらワークメモリ103に記憶するまた、ステップS31で、ワークメモリ103にフォーム情報が記憶されていた場合にはステップS32に進み、ワークメモリ103のフォーム情報を伸長しながら、ビットマップメモリ105に展開する。その後ステップS36に進み、一般的な文書データ(このフォーム情報に重ね合わされる)をビットマップメモリ105に展開し、フォーム情報と文書データとが重なった状態で完成したビットマップメモリ105のイメージデータをプリンタエンジン部107に送出することにより印刷を行う(ステップS37)。

【0035】尚、上述した予想圧縮データサイズの算出方法は、圧縮の方法によっても異なるため、これを最適化することにより、より一層の印刷処理を高速化できる。

【0036】さらに、圧縮転送処理、伸長転送処理を、他の制御処理で用いられているような、ハードウェアで達成することにより、資源の有効活用、そして、より一層の高速化が可能となる。

【0037】次に本発明の第4実施例を説明する。尚、この第4実施例のハードウェアの構成は、前述の図4と同様であるため、これらの説明を省略する。

【0038】この第4実施例では、キャッシュメモリ領域に記憶する文字パターンデータに優先順位を設け、優先順位の低いものから削除するよう構成することにより、キャッシュメモリ領域が一杯になった時でも、蓄積した文字パターンが一度に削除されることがなくなる。これにより、高い頻度で使用する文字パターンがキャッシュメモリに保存されている確率が高くなり、文字コードよりフォントメモリを参照してパターン展開することなく、既に展開されたパターンをキャッシュメモリより取り出して利用できる確率を高めたものである。

【0039】図7は第4実施例におけるキャッシュメモリ106aの構成を示す図である。図7において、801はキャッシュ管理テーブル、802はドットパターン

に展開されたパターンデータを記憶するためのパターンデータ領域を示している。

【0040】また、図8は、キャッシュ管理テーブル801に記憶されている1バイトの登録文字パターンの優先順位情報を示している。この優先順位は、対応する属性のビットをオン“1”にすることにより、優先順位情報を記憶している。この優先順位はビット0(現在のページで使用中の文字セット)が最も優先順位が高く、ビット7(40以上の大きな文字)が最も優先順位が低くなっている。

【0041】また、キャッシュ管理テーブル801には、選択利用されるための文字属性情報の他に、図8に示す優先順位情報、更にはパターンデータのアドレスを記憶しており、文字データの印刷の際にキャッシュ管理テーブル801を文字属性によって検索し、該当する文字がキャッシュメモリ106aに登録されていれば、パターンアドレス部によって指定されているパターンデータを、パターンデータ領域802より読出して印刷する。

【0042】図9及び図10は、本発明の第4実施例におけるCPU101の文字データ処理手順を説明するためのフローチャートで、ここでも前述のフローチャートの場合と同様に、ホストコンピュータ300よりのデータの受信が、この処理と並行して行われており、受信バッファ108を介してページバッファメモリ104に受信データが順次格納されている。

【0043】まずステップS41で、ページバッファメモリ104より1文字分のデータを読出す。次にステップS42で、キャッシュメモリ106aのキャッシュ管理テーブル801を参照して登録文字を検索し、その読出した文字がキャッシュメモリ106aに記憶されているかどうかを判断する。その文字データがキャッシュメモリ106aに登録されている時はステップS43に進み、キャッシュメモリ106aのパターンデータ領域802から該当する文字パターンを読出し、ステップS61でビットマップメモリ105に展開して、1文字のパターン展開を終了する。

【0044】一方、ステップS42において、その文字パターンがキャッシュメモリ106aに登録されていない時はステップS44に進み、該当するアウトラインデータのフォント情報をフォントメモリ102より読み込み、そのデータに基づいて文字パターンを発生させた際にキャッシュメモリ106aの登録スペース(メモリ容量)が足りるかどうかを判断する(ステップS45)。ステップS45でキャッシュメモリ106aの残量が多い時はステップS59に進み、その文字をパターン展開し、ステップS60でキャッシュメモリ106aにその文字パターンを登録してステップS61に進み、ビットマップに展開する。

【0045】ステップS45で、キャッシュメモリ10

6 a のメモリ容量が不足している時はステップ S 4 6 に進み、キャッシュ管理テーブル 8 0 1 の優先順位情報を参照して、キャッシュメモリ 1 0 6 a 内の優先順位の低い情報から削除していく。まず、ステップ S 4 6 において、現在使用されておらず、かつアサイン番号（頻繁に使用される文字セットにユーザが割りつける番号）が割り付けられていない文字セットの内、文字のサイズが大きなものについてキャッシュメモリ 1 0 6 a より削除する。この処理の後、ステップ S 4 7 で、文字パターンを発生させた際にキャッシュメモリ 1 0 6 a の登録スペース（メモリ容量）が足りるようになったかどうかを判断し、足りた場合はステップ S 5 9 に進み、前述のように文字パターンの展開、キャッシュメモリ 1 0 6 a への登録等を行う。

【0 0 4 6】ステップ S 4 7 の判断が否であればステップ S 4 8 に進み、現在使用されておらず、かつアサイン番号が割り付けられていない文字セットの内、修飾された文字を削除し、ステップ S 4 6 と同様にキャッシュメモリ 1 0 6 a に登録できるだけのスペースができたかどうかを判断する。そして、以下ステップ S 5 0 ～ステップ S 5 5 において、図 8 に示す優先順位情報における優先順位の低い順に第 2 水準の漢字、次に第 1 水準の漢字を削除し（ステップ S 5 0）、ステップ S 5 3 ではカタカナ文字を、ステップ S 5 4 では平仮名文字を削除する。そして、これら各文字種を削除する度に、キャッシュメモリ 1 0 6 a に登録スペースが確保できたかを調べる。

【0 0 4 7】このようにして、どうしても登録スペースが足りない時はステップ S 5 6 に進み、現在のページで使用されていない文字を全てキャッシュメモリ 1 0 6 a から削除し、ステップ S 5 7 でどうしても登録スペースが足りない時はステップ S 5 8 で、キャッシュメモリ 1 0 6 a への登録を行わず、ワーク領域を使用して展開し、ビットマップメモリ 1 0 5 に展開する。

【0 0 4 8】尚、上記実施例において、段階分けを行った優先順位については、プリンタの用途、機能によって異なり、プリンタによって最適化を行うことによって、より一層の効果を得ることができる。

【0 0 4 9】また、優先順位情報を 1 ビットとして持つのではなく、展開時間を段階的に記憶しておく数ビットの情報としたり、頻度情報として使用回数をカウントしたり、定期的に削除するための指標として、使用されなくなって何ページ出力したかをカウントする等のカウンタとして持つことも考えられる。

【0 0 5 0】尚、前述の第 4 実施例では、優先順位は固定であるとして説明したが、ユーザがパネルやコマンド等で優先順位を任意に設定できるように構成することにより、ユーザのニーズに合ったキャッシュメモリ 1 0 6 a のパターンデータの制御を行うことができる。例えば、キャッシュメモリの文字パターンの優先順位設定コ

マンドを設け、図 8 に示した優先順位情報内の上位ビット（ビット 7 の 4 0 以上の大きな文字）から、順に数値をコマンドで設定することにより、数値が大きい順に高く変更される。

【0 0 5 1】また、ユーザが設定したキャッシュメモリ 1 0 6 a のパターンデータの優先順位を、電源を切って再度立ち上げた場合に設定し直さなくても良いように、プリンタに不揮発性の記憶エリアを設け、優先順位を表す数値を記憶しておくことによりユーザフレンドリーなプリンタとなることが可能になる。

【0 0 5 2】また、前述の実施例として、キャッシュメモリ 1 0 6 a のパターンデータを削除する際の基準としてのみ優先順位を使用したのが、本発明はこれに限定されず、例えば文字パターンの展開に時間がかかる文字を予め展開しておくように優先順位情報を設定する等、他の処理に利用することもできる。

【0 0 5 3】以上説明したように、この第 4 実施例によれば、キャッシュメモリを有効に活用することができ、高頻度で使用する文字パターンを再度パターン展開することなく、キャッシュメモリから取り出してビットマップに展開できるため、印刷速度を向上できる。

【0 0 5 4】尚、本発明は複数の機器から構成されるシステムに適用しても、1 つの機器からなる装置に適用しても良い。また、本発明はシステム或は装置に、本発明を実施するプログラムを供給することによって達成される場合にも適用できることはいうまでもない。

【0 0 5 5】

【発明の効果】以上説明した様に本発明によれば、フォーム情報を圧縮してキャッシュとして記憶しておき、フォームオーバーレイが指示された時に、その他の画像データと記憶されているフォーム情報とを合成して画像を形成することにより、高速にフォームオーバーレイ画像を形成できる効果がある。

【0 0 5 6】更に、他の発明によれば、キャッシュメモリの登録データに優先順位を持たせ、キャッシュメモリの残量が少なくなると、その優先順位の低い順に登録されているデータを消去するようにしてキャッシュメモリの使用効率を高めることができる。

【図面の簡単な説明】

【図 1】本発明の第 1 実施例の印刷装置の概略構成を示すブロック図である。

【図 2】第 1 実施例におけるオーバーレイ情報のサンプルであるフォーム情報と一般データである文書データの一例を示す図である。

【図 3】本発明の第 1 実施例における CPU の動作を示すフローチャートである。

【図 4】本発明の第 2 実施例における印刷装置の概略構成を示すブロック図である。

【図 5】本発明の第 2 実施例における CPU の動作を示すフローチャートである。

【図6】本発明の第3実施例におけるCPUの動作を示すフローチャートである。

【図7】本発明の第4実施例の印刷装置におけるキャッシュメモリのデータの格納状態を示す図である。

【図8】第4実施例のキャッシュ管理テーブルの優先順位情報を説明するための図である。

【図9】本発明の第4実施例の印刷装置におけるCPUの動作を示すフローチャートである。

【図10】本発明の第4実施例の印刷装置におけるCPUの動作を示すフローチャートである。

【符号の説明】

100 印刷装置

101 CPU

102 フォントメモリ

103 ワークメモリ

104 ページバッファメモリ

105 ビットマップメモリ

106, 106a キャッシュメモリ

107 プリンタエンジン部

108 受信バッファ

200 印刷物

201 フォーム情報

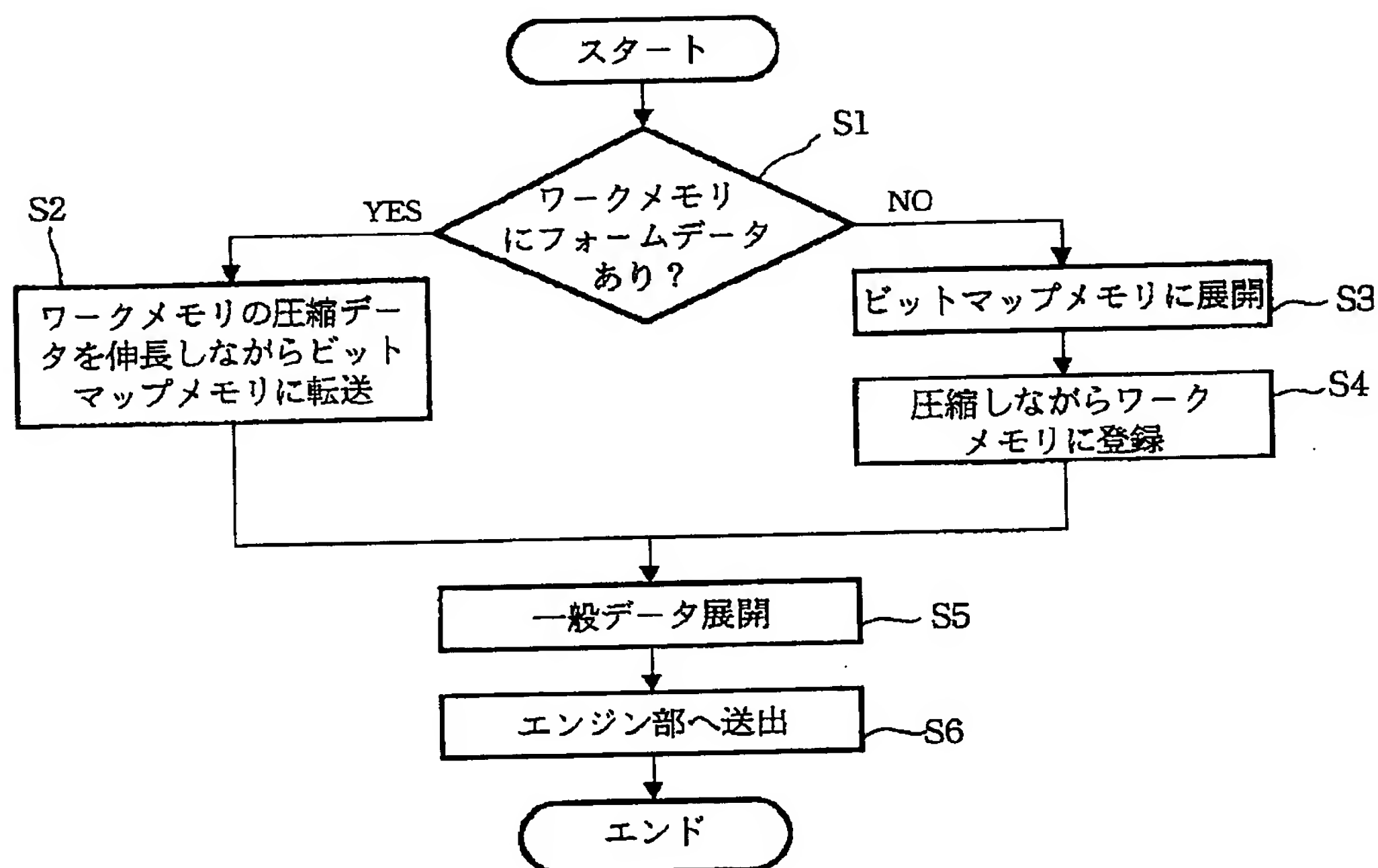
202 文書データ (一般データ)

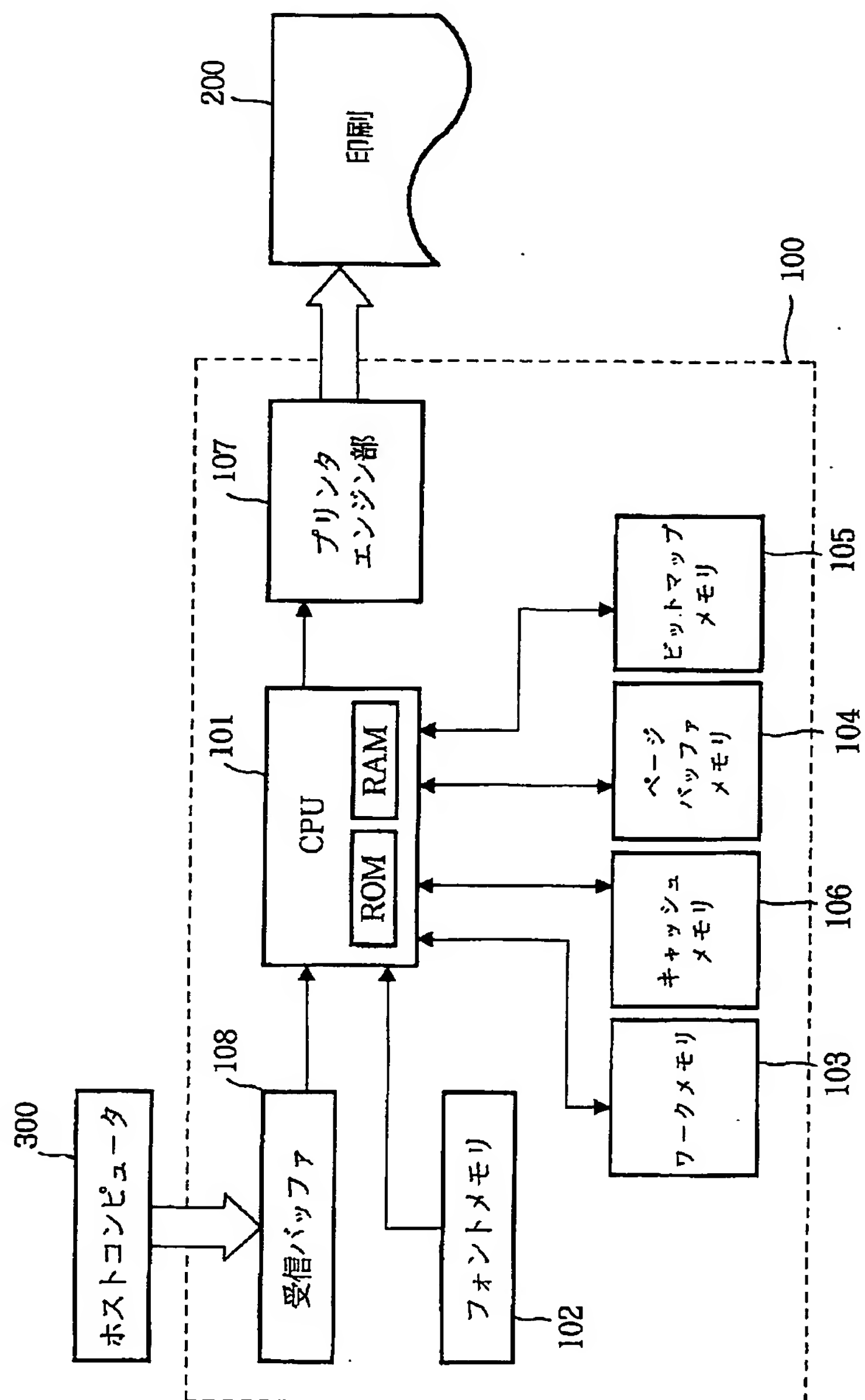
300 ホストコンピュータ

801 キャッシュ管理テーブル

802 キャッシュデータ

【図3】





【図1】

【図2】

フォーム

カテゴリー価格 ()

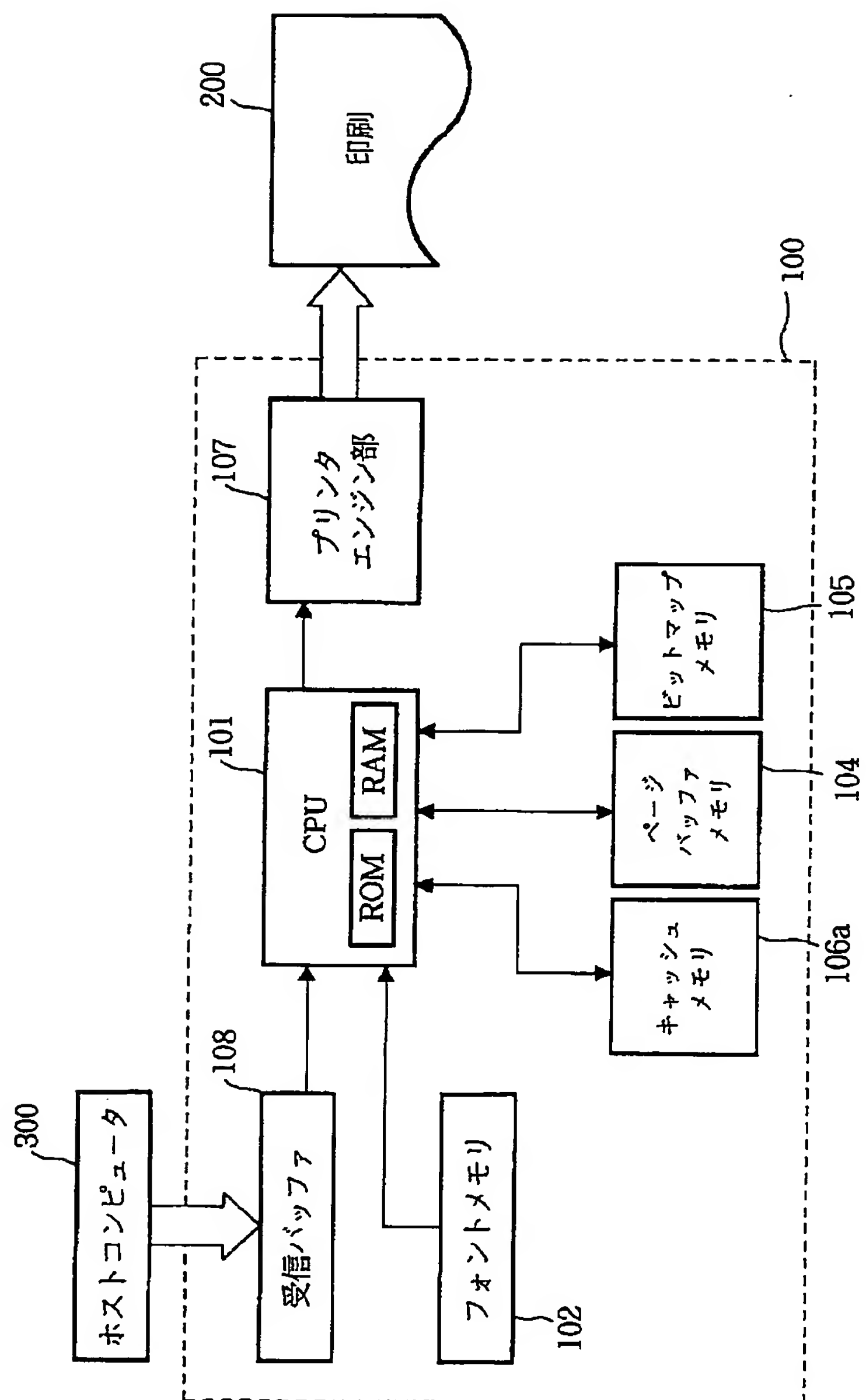
品名	単価	最低個数
えんぴつ	30	12
パイスタA	400	5
パイスタB	600	5
ボールペン	100	12
消しゴム	50	6
定規	200	1
ノートA	100	6
ノートB	200	6
封筒A	60	12
封筒B	100	12
封筒C	200	12

一般データ

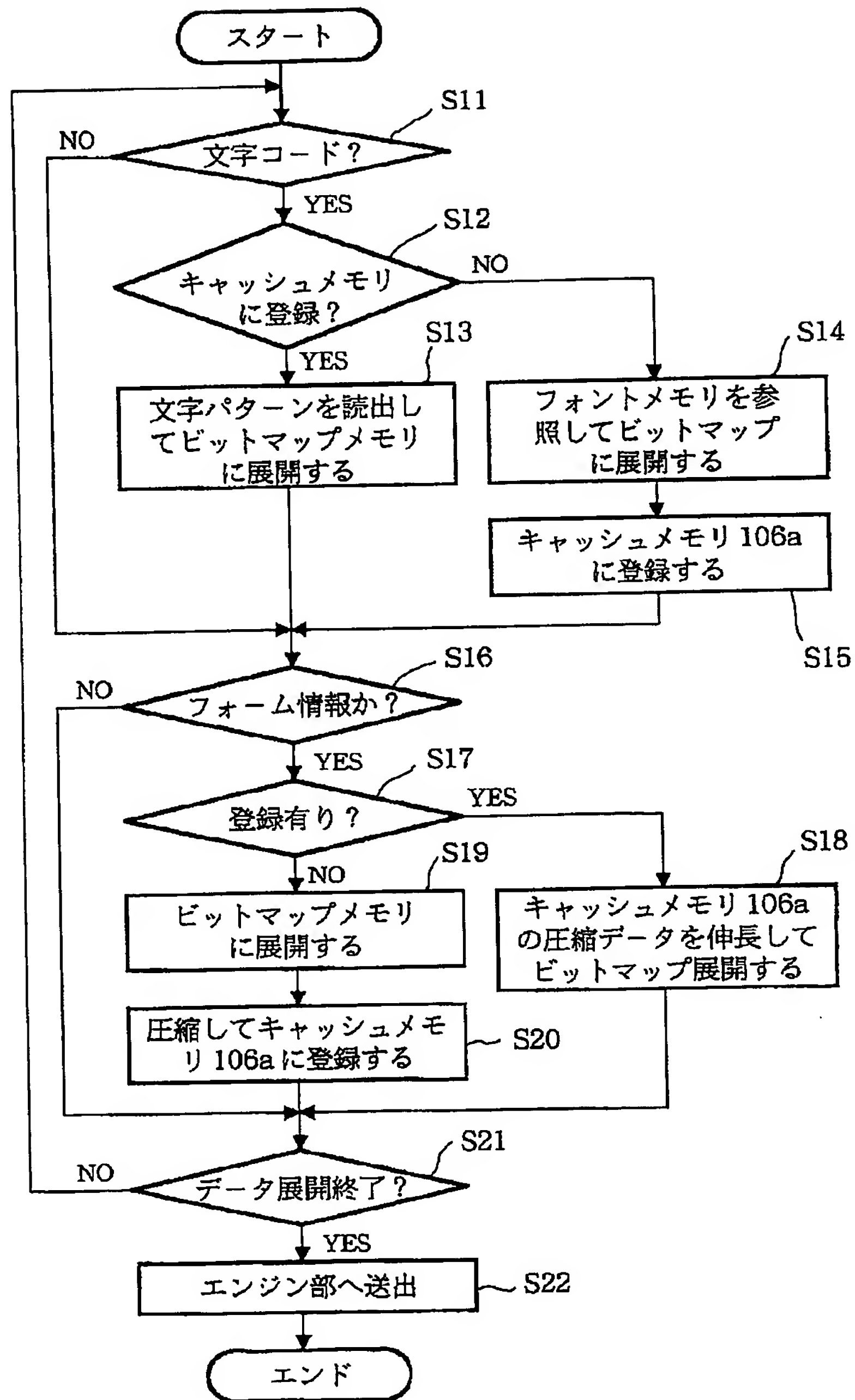
1/3

えんぴつ	30	12
パイスタA	400	5
パイスタB	600	5
ボールペン	100	12
消しゴム	50	6
定規	200	1
ノートA	100	6
ノートB	200	6
封筒A	60	12
封筒B	100	12
封筒C	200	12

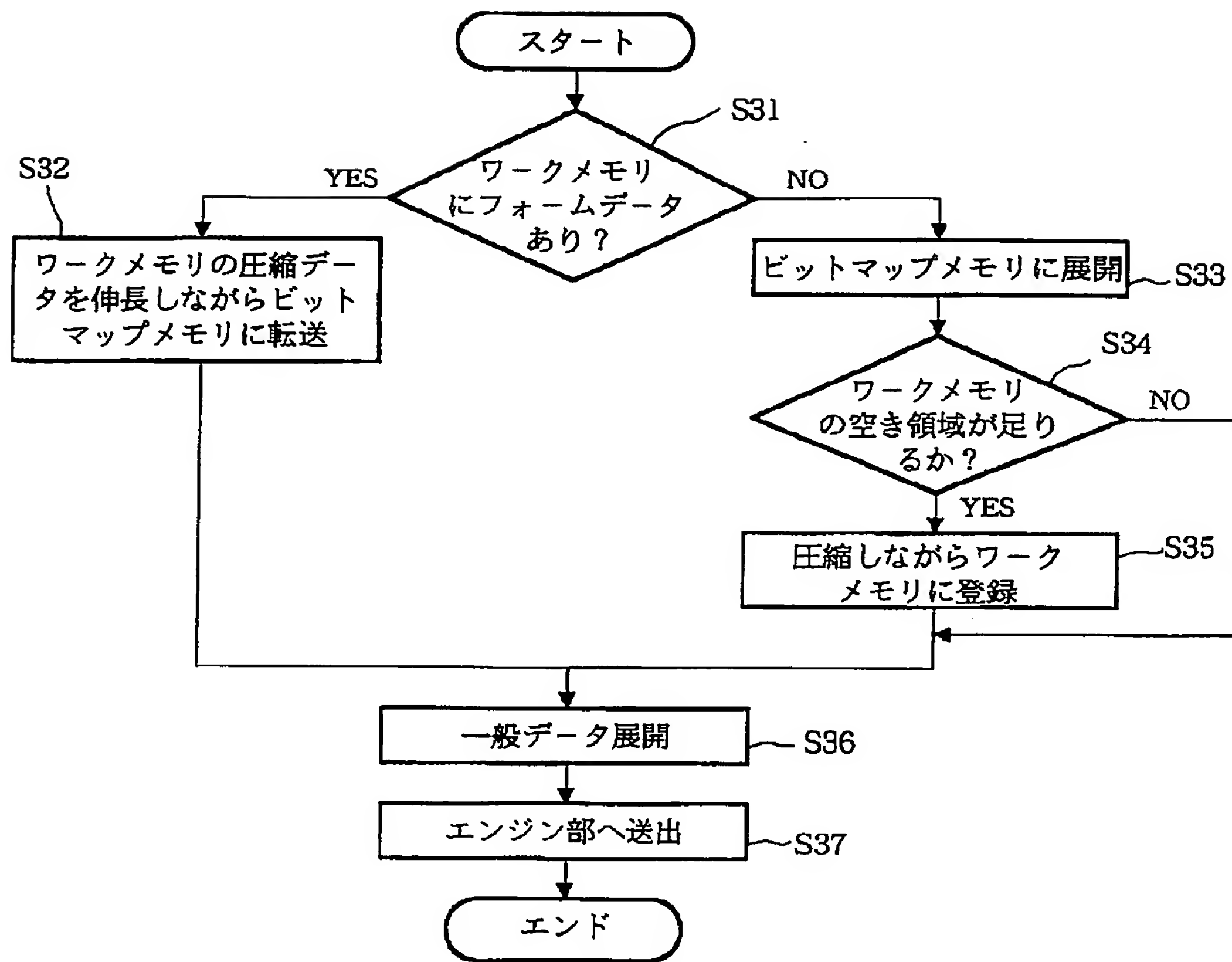
【図4】



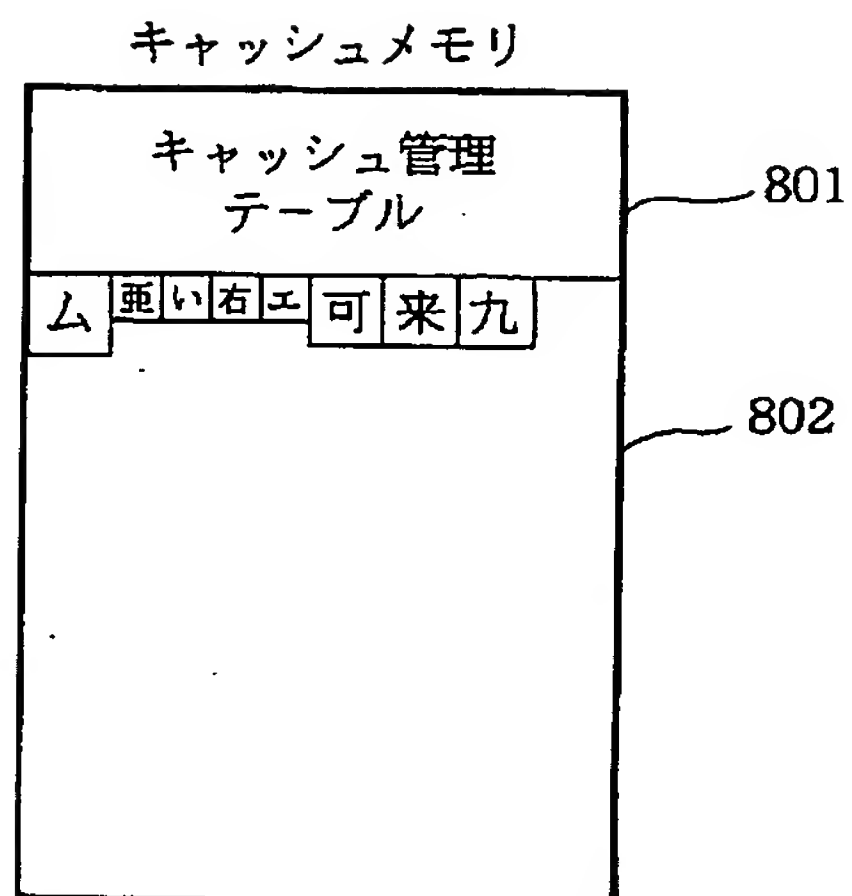
【図5】



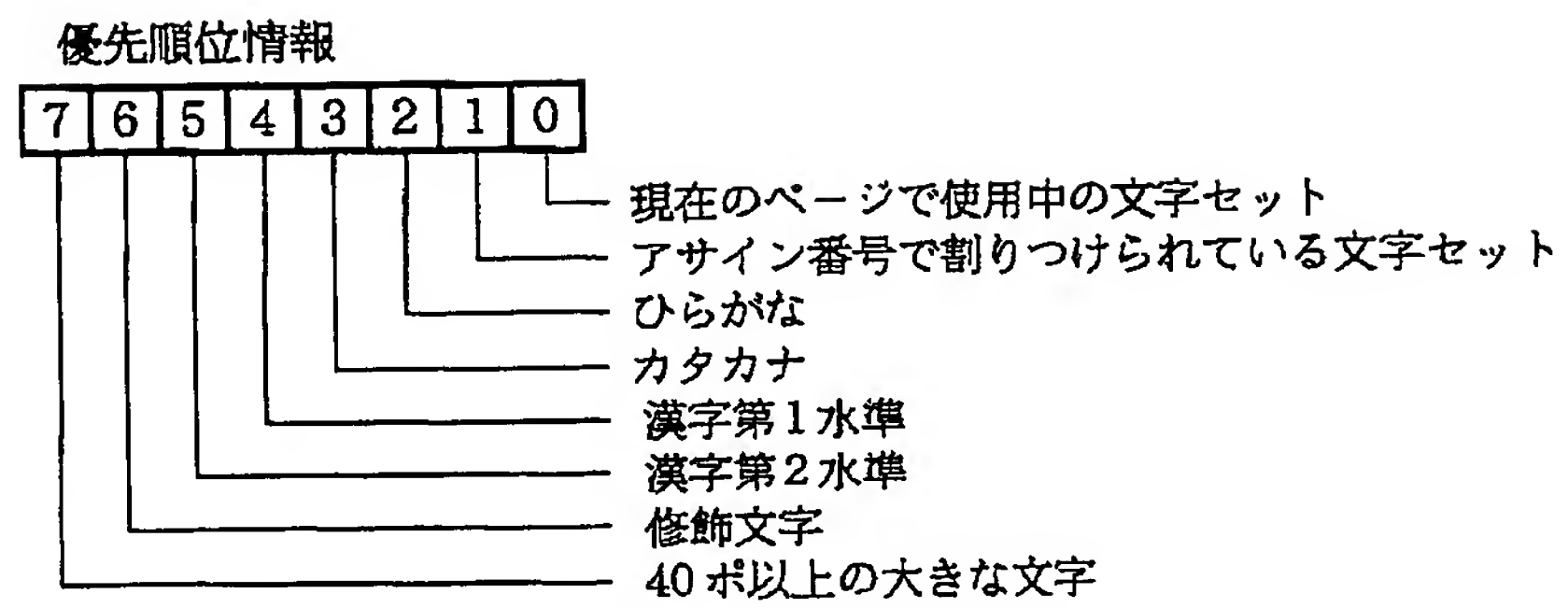
【図6】



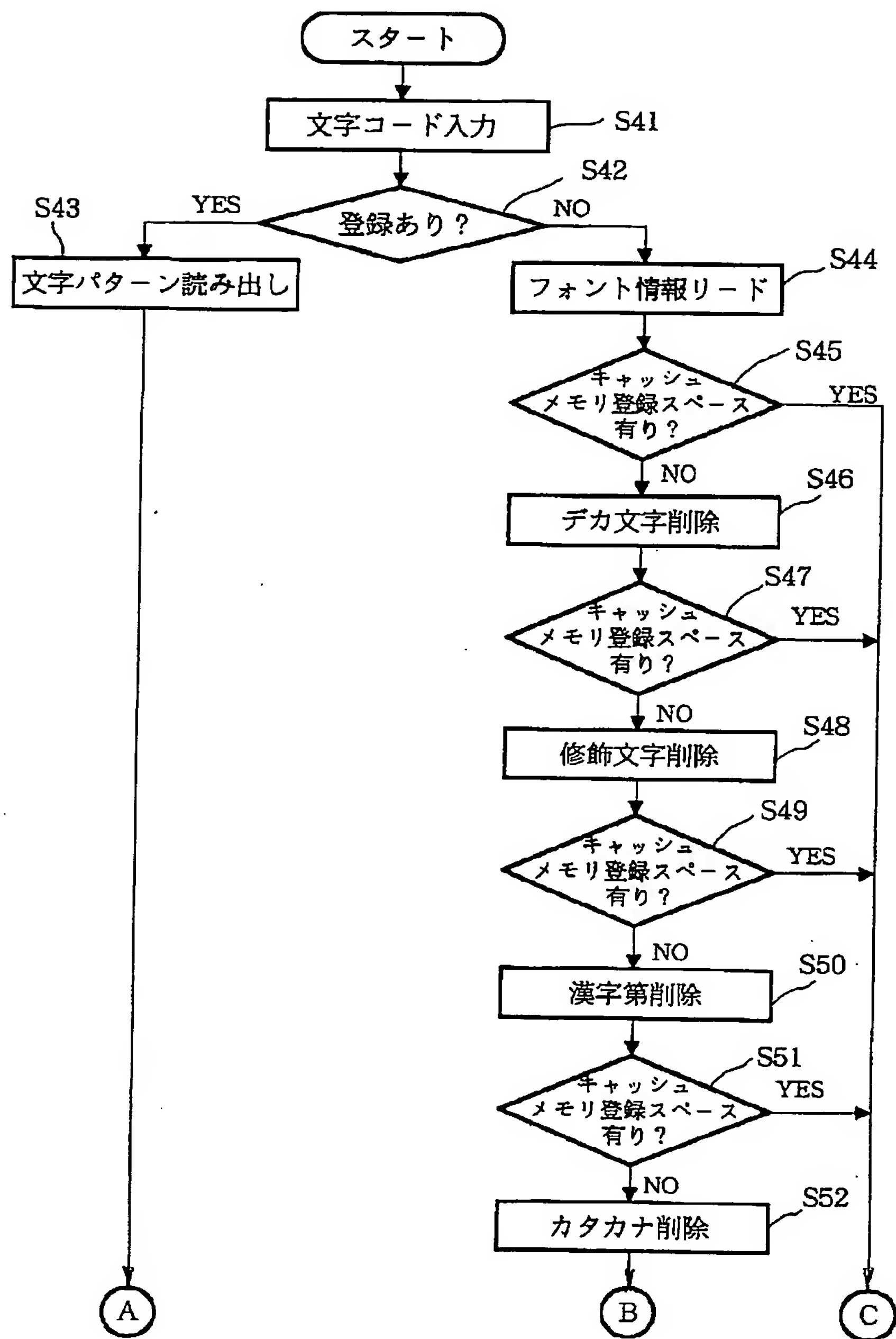
【図7】



【図8】



【図9】



【図10】

